**MAC 747 GEOMETRIA COMPUTACIONAL**

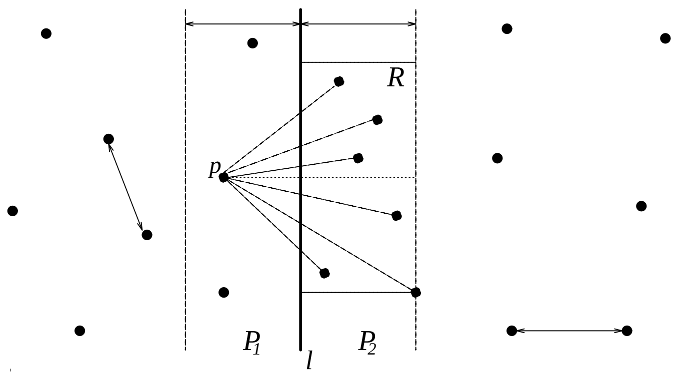
LISTA DE EXERCÍCIOS 1

## 1. Instruções

Entregue os exercícios 2, 4 e 5 até as 17:00 da sexta-feira dia 22 de agosto na secretaria do Departamento de Ciência da Computação (sala 256 do Bloco A).

## 2. Exercícios

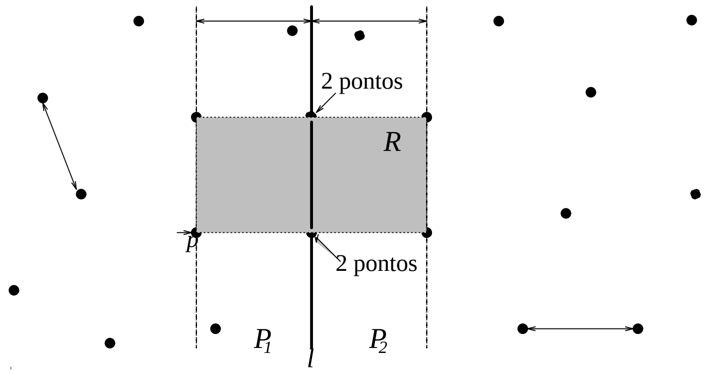
1. Considere a fase de COMBINAR do algoritmo de divisão-e-conquista para o Problema do Par Mais Próximo. Nesta fase, para cada ponto $p \in P_1$ o algoritmo calcula a distância entre $p$ e no máximo 6 pontos em $P_2$. Você consegue mostrar um exemplo onde é realmente **necessário** calcularmos a distância entre $p$ e 6 pontos $q_1, \ldots, q_6$ de $P_2$? (Ou seja, o seu exemplo deve mostrar um ponto $p$ em $P_1$ e pontos $q_1, \ldots, q_6$ contidos no retângulo $R$ de tal forma que se o algoritmo calcular a distância entre $p$ e cada um dos pontos (digamos) $q_1, \ldots, q_5$ então o algoritmo não devolverá como resposta o par mais próximo (que por azar é o par $\{p, q_6\}$.) Se você não conseguir encontrar um tal exemplo, então tente mostrar que é **suficiente** o algoritmo calcular a distância entre cada ponto de $P_1$ e menos do que 6 pontos de $P_2$.




2. Considere agora a seguinte pequena variação da fase COMBINAR do algoritmo. Seja $P^*$ a lista dos pontos em $P_1 \cup P_2$ ordenados em relação as suas $y$-coordenadas (abcissas). Percorra a lista $P^*$ em ordem e para cada ponto $p \in P^*$ calcule a distância entre $p$ e os 7 pontos que seguem $p$ na lista $P^*$. (Note que algumas vezes poderemos estar calculando a distância entre pontos que estão no mesmo conjunto $P_i$ (i=1,2).) Mostre que com esta variação da fase COMBINAR o algoritmo esta correto (i.e. encontra o par mais próximo).

[Note que com esta variação não precisamos mais ficar testando as $y$-coordenadas dos pontos que terão as suas distâncias calculadas.

Sugestão: Veja a figura abaixo ou de uma olhada em [1] página 910.]




3. Considere ainda a variação da fase COMBINAR proposta no exercício anterior. Você mostrou que é **suficiente** calcularmos as distâncias entre cada ponto $p \in P^*$ e os 7 pontos que seguem $p$ na lista $P^*$ para que o algoritmo esteja correto. Será que é **necessário** calcularmos essas 7 distâncias? Ou seja, será que não poderiamos calcular a distância entre cada ponto $p \in P^*$ e os (digamos) 6 pontos que seguem $p$ na lista $P^*$ e ainda assim o algoritmo estaria correto? Apresente um exemplo mostrando que é necessário calcularmos essas 7 distâncias ou mostre que é possível encontrarmos o par mais próximo calculando menos do que 7 distâncias para cada $p \in P^*$.

   [Note que para a ordem de complexidade do algoritmo tanto faz se para cada $p \in P^*$ calculamos 7, 6 ou 100000 distâncias—o algoritmo continua tendo complexidade de tempo $O(n \log n)$.]
4. (Exercício 33.4-3 de [1]) A distância entre dois pontos pode ser definida de diversas maneiras além da Euclidiana. No plano, a $L_m$-distância entre dois pontos $p_1 = (x_1, y_1)$ e $p_2 = (x_2, y_2)$ é dada por $(|x_1 - x_2|^m + |y_1 - y_2|^m)^{1/m}$. Portanto a distância Euclidiana é a $L_2$-distância. Modifique o algoritmo de divisão-e-conquista para o Problema do Par Mais Próximo de tal forma que ele devolva o par de pontos mais próximos em relação a $L_1$-distância (também conhecida como *Manhattan distance*).
5. (Guardando as paredes) Construa um polígono $P$ e coloque guardas em $P$ de tal forma que cada ponto em $\partial P$ seja visto por pelo menos um guarda, mas exista algum ponto de $P$ que não é visto por nenhum dos guardas.

## Referências

1. T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to algorithms*, The MIT Electrical Engineering and Computer Scienece Series, The MIT Press, MacGraw-Hill Book Company, 1990, QA758 C811i.